REMARKS

This Application has been carefully reviewed in light of the Advisory Action mailed May 10, 2006 ("Office Action"). At the time of the Advisory Action, Claims 1-21 were pending in the application. To advance prosecution of this case, Applicant amends Claims 1-3, 5-11, 13-16, 18-19, and 21. In addition, Applicant cancels Claims 4 and 17. Applicant does not admit that any amendments are due to any prior art or any of the Examiner's rejections. Applicant respectfully requests reconsideration and allowance of all pending claims.

Claim Rejections - 35 U.S.C. § 102

The Examiner rejects Claims 1-21 under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,826,013 issued to Nachenberg (*"Nachenberg"*). Applicant respectfully requests reconsideration and allowance of Claims 1-21.

Nachenberg fails to support the rejection for several reasons. First, the cited reference fails to teach, suggest, or disclose "detecting at least one modification to a memory state of a computer system, wherein...the at least one modification...comprises installation of a viral interrupt handler...and associates the particular interrupt with the viral interrupt handler instead of the legitimate interrupt handler" as recited, in part, in amended Claim 5. Second, the cited reference fails to teach, suggest, or disclose "detecting at least one modification to a memory state of a computer system, wherein the at least one modification...comprises insertion of a pointer to a viral exception handler, the pointer associated with a particular exception" as recited, in part, in amended Claim 1. Third, *Nachenberg* fails to teach, suggest, or disclose "detecting at least one instruction, wherein the at least one instruction forces the particular exception" as recited, in part, in amended Claim 1.

First, the cited reference fails to teach, suggest, or disclose "detecting at least one modification to a memory state of a computer system, wherein...the at least one modification...comprises installation of a viral interrupt handler...and associates the particular interrupt with the viral interrupt handler instead of the legitimate interrupt handler" as recited, in part, in amended Claim 5. *Nachenberg* discloses a virus detection method that involves emulating computer instructions and comparing the emulated instructions to instruction/interrupt profiles of known viruses. (*Nachenberg*; col. 3, ll. 37-46). In the Final Office Action dated January 11, 2006, the Examiner cites the following portion of

Nachenberg:

> The dynamic exclusion module (240) examines the instruction/interrupt usage profiles (224) of each known polymorphic virus (150) as each instruction is fetched for emulation. The instruction/interrupt usage profiles (224) indicate which polymorphic viruses (150) employ mutation engines that do not use the fetched instruction in decryption loops they generate, and the emulation control module (220) flags these viruses. The emulation control module (220) continues until all mutation engines have been flagged or until a threshold number of instructions have been emulated. The flagging technique implemented by the dynamic exclusion module (240) determines when emulation has proceeded to a point where at least some code from the decrypted static virus body (160) may be scanned and substantially reduces the number of instructions emulated prior to scanning the remaining target files without resort to booster or stopper heuristics.
>
> ...
>
> It is not always necessary to fully decrypt the static virus body (160) to identify the underlying virus. In the preferred embodiment of the invention, the emulation control module (220) tracks those parts of virtual memory modified during emulation and periodically interrupts the emulation process to call the scanning module (250). The scanning module (250) tries to identify the virus type from the portion of decrypted static virus code (160). In order to speed up the process, the scanning module (250) implements a coarse scan of tagged memory locations to identify data bytes most likely to be associated with decrypted static virus code (virus signatures). It implements a more detailed binary search process only when selected bytes are encountered during the coarse scan. This approach greatly speeds up scanning without decreasing the accuracy of the scanning module (250). When code matching one of the viral signatures is identified, the PAM system (200) signals to the host computer that an infected file has been located.

(*Nachenberg*; col. 3, ll. 37-46; col. 4, ll. 24-31). Thus, *Nachenberg* discloses a method for emulating computer instructions. In particular, *Nachenberg* describes a module that "tracks those parts of virtual memory modified during emulation and periodically interrupts the emulation process to call the scanning module." (*Nachenberg*; col. 4, ll. 25-29). Merely interrupting an emulation process, however, has nothing to do with "installation of a viral interrupt handler" as recited, in part, in amended Claim 5. In addition, while *Nachenberg* mentions an "instruction/interrupt usage profile," such a profile does not teach, suggest, or disclose a modification that "associates the particular interrupt with the viral interrupt handler instead of the legitimate interrupt handler" as recited in amended Claim 5. Indeed, nothing in the cited reference teaches, suggests, or discloses a "legitimate interrupt handler" or a "viral interrupt handler" as recited in amended Claim 5.

The Examiner's reliance on another portion of *Nachenberg* is similarly flawed. In the Office Action, the Examiner relies on a portion of *Nachenberg* that describes detecting "non-initialized indexed writes" and determining whether an "index register has been initialized or modified." (*Nachenberg*; col. 12, ll. 65-66; col. 13, ll. 3-5). In the Final Office Action, the Examiner seems to equate the non-initialized indexed writes and the modified index register in *Nachenberg* with "the at least one modification" recited in various claims. Notably, however, *Nachenberg* fails to teach, suggest, or disclose that the non-initialized indexed writes and the modified index registers comprise installation of anything. (*Nachenberg*; col. 12, ll. 65-66; col. 13, ll. 3-5). Thus, the cited reference fails to teach, suggest, or disclose that "the at least one modification...comprises installation of a viral interrupt handler" as recited, in part, in Claim 5. It is well established that a "claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. Union Oil Co. of California*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987). In addition, "[t]he identical invention *must* be shown in as complete detail as is contained in the...claim," and "[t]he elements *must* be arranged as required by the claim." *Richardson v. Suzuki Motor Co.*, 868 F.2d 1226, 1236, 9 U.S.P.Q.2d 1913, 1920 (Fed. Cir. 1989); *In re Bond*, 15 U.S.P.Q.2d 1566 (Fed. Cir. 1990); MPEP § 2131 (emphasis added). The cited reference fails to teach, suggest, or disclose "detecting at least one modification to a memory state of a computer system, wherein...the at least one modification...comprises installation of a viral interrupt handler...and associates the particular interrupt with the viral interrupt handler instead of the legitimate interrupt handler" as recited, in part, in amended Claim 5. Accordingly, amended Claim 5 is not anticipated by *Nachenberg*.

Second, the cited reference fails to teach, suggest, or disclose "detecting at least one modification to a memory state of a computer system, wherein the at least one modification...comprises insertion of a pointer to a viral exception handler, the pointer associated with a particular exception" as recited, in part, in amended Claim 1. As explained above, the Office Action relies on a portion of *Nachenberg* that discloses (1) tracking memory that is modified during emulation of code, and (2) periodically interrupting the emulation to call a scanning module that "tries to identify the virus type from the portion of decrytped static virus code." (*Nachenberg*; col. 4, ll. 25-32). Merely tracking memory and calling a scanning module, however, does not teach, suggest, or disclose that the modification "comprises insertion of a pointer to a viral exception handler" as recited in amended Claim 1. Indeed, there is nothing in *Nachenberg* that teaches, suggests, or discloses "a viral exception handler" as recited in amended Claim 1. Because the cited reference fails to teach, suggest, or disclose these elements of amended Claim 1, the cited reference fails to support the rejection.

Third, *Nachenberg* fails to teach, suggest, or disclose "detecting at least one instruction, wherein the at least one instruction forces the particular exception" as recited, in part, in amended Claim 1. Notably, amended Claim 1 recites that "the particular exception" is associated with the "pointer to a viral exception handler." There is nothing in *Nachenberg* that teaches, suggests, or discloses an instruction that "forces the particular exception" that is associated with the "pointer to a viral exception handler," as recited in amended Claim 1. Accordingly, *Nachenberg* fails to support the rejection.

In rejecting Claims 8-11, the Examiner employs the same rationale used with respect to Claims 1 and 5. Accordingly, for at least the reasons stated above, Applicant respectfully requests reconsideration and allowance of amended Claims 8-11.

Claims 2-3, 6-7, 12-16, 18-21 depend from independent claims shown above to be allowable. In addition, these claims recite further elements not taught, suggested, or disclosed by the cited reference. Accordingly, Applicant respectfully requests reconsideration and allowance of Claims 2-3, 6-7, 12-16, 18-21.

CONCLUSION

Applicant has made an earnest attempt to place this case in condition for allowance. For the foregoing reasons, and for other reasons clearly apparent, Applicant respectfully requests full allowance of all pending claims.

If the Examiner feels that a telephone conference would advance prosecution of this Application in any manner, the Examiner is invited to contact Samir A. Bhavsar, Attorney for Applicant, at the Examiner's convenience at (214) 953-6581.

The Commissioner is hereby authorized to charge any fees or credit any overpayment to Deposit Account No. 02-0384 of Baker Botts L.L.P.

Respectfully submitted,

BAKER BOTTS L.L.P.
Attorneys for Applicant

Samir A. Bhavsar
Reg. No. 41,617

Date: June 12, 2006

CORRESPONDENCE ADDRESS:

at Customer No. **05073**